

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине
‘Функциональная схемотехника’

Вариант №6

Выполнил:
Студент группы Р33312
Соболев Иван
Александрович
Преподаватель:
Табунщик Сергей
Михайлович



Санкт-Петербург, 2024

Оглавление

Цели работы.....	2
Задание.	2
Схема разработанного блока вычисления функции.	2
Описание работы разработанного блока, начиная с подачи входных данных и заканчивая получением результата.....	3
Область допустимых значений для разработанного блока.	4
Результат тестирования разработанного блока.	5
Временная диаграмма тестирования.....	6
Время вычисления результата при частоте тактового сигнала в 100 МГц.	6
Выводы.	7

Цели работы.

Получить навыки описания арифметических блоков на RTL-уровне с использованием языка описания аппаратуры Verilog HDL.

Задание.

6	$y = a \cdot b + a^3$	2 сумматора и 1 умножитель
---	-----------------------	----------------------------

Рисунок 1 – Задание

Схема разработанного блока вычисления функции.

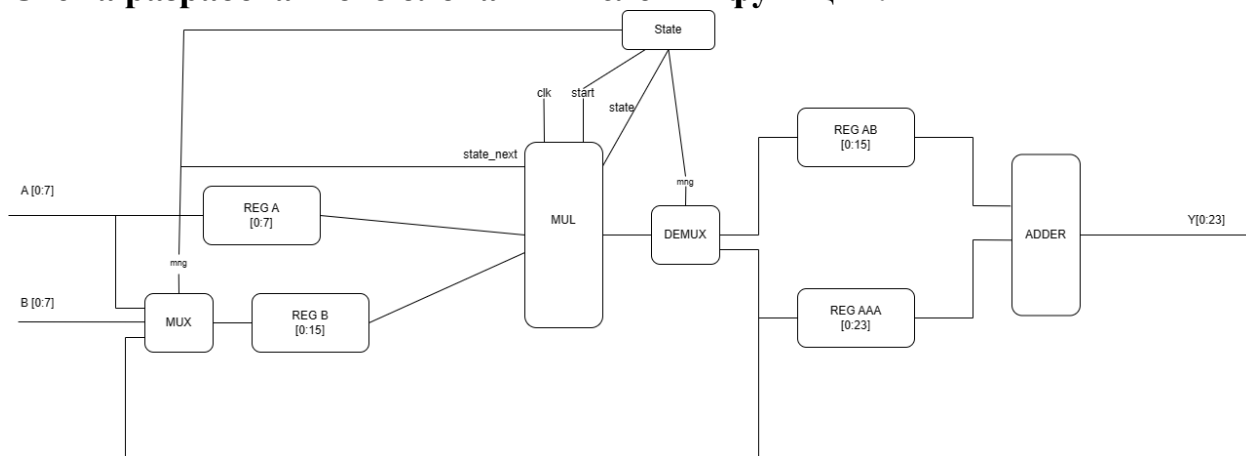


Рисунок 2 - схема разработанного блока

В схеме используется мультиплексор и демультимплексор. Мультиплексор — это устройство, которое позволяет выбирать один из нескольких входных сигналов и перенаправлять его на выход. У мультиплексора имеется один выход и несколько входов для выбора данных. Демультимплексор принимает на вход один общий сигнал и перенаправляет его на нужный вход. Управляющие сигналы определяют, на какой из входов будет перенаправлен общий входной сигнал.

Для реализации данной функции был разработан и протестирован блок умножения.

Временная диаграмма тестирования блока умножения:

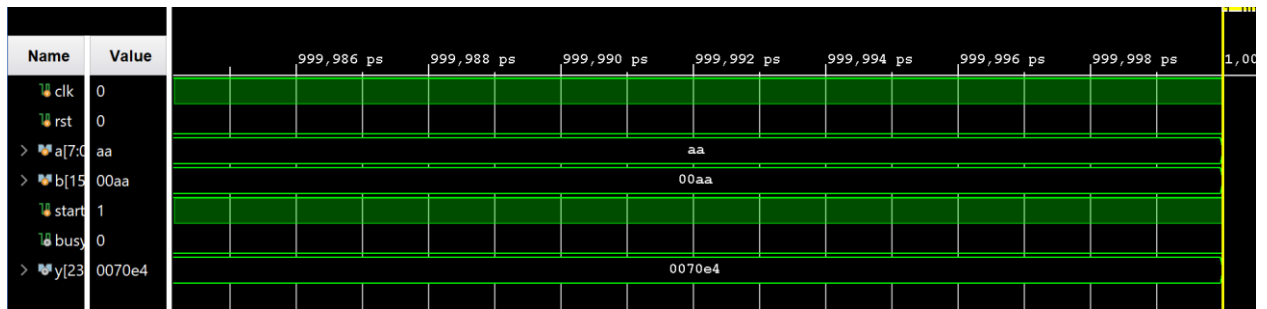


Рисунок 3 - временная диаграмма блока MUL

Перемножение двух шестнадцатеричных чисел AA (в десятичном виде 170) даёт результат 70E4 (в десятичном виде 28900), что является верным результатом.

Для реализации функции был также разработан конечный автомат Мура:

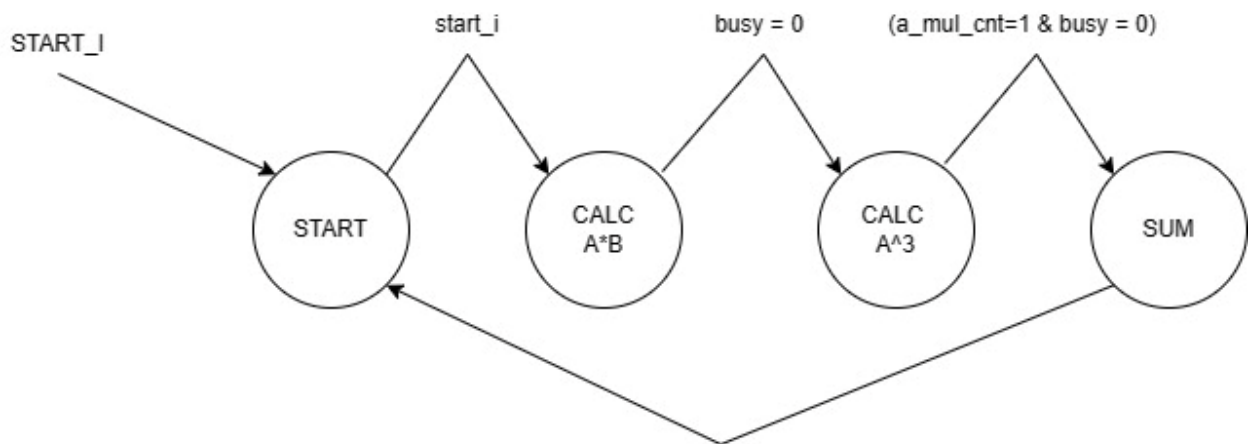


Рисунок 4 - автомат Мура

Описание работы разработанного блока, начиная с подачи входных данных и заканчивая получением результата.

На вход моего блока подаются сигналы:

- Start - сигнал, говорящий о начале вычислений
- Rst - сигнал сброса
- Clk - вход синхроимпульсов
- A - первый аргумент
- B - второй аргумент

Сам алгоритм подсчета можно разбить на данные следующие этапы:

- Модуль ожидает, когда во вход start будет подана 1
- Как только на вход подается сигнал о начале работы модуля (start), то с входных данных A B записываются соответствующие значения в регистры A B. Далее на

вход умножителя подаются два операнда А, В. Результат умножения записывается в регистр АВ.

- На вход регистра В, подается значение с регистра А, и на вход умножителя подаются два значения А, результат умножения приходит в регистр В, через демультимплексор.
- После этого мы последний раз подаем на вход умножителя значения с регистров и результат записывается в регистр ААА.
- Как только сформированы оба операнда для сложения мы складываем их и подаем результат на выход.

Область допустимых значений для разработанного блока.

Так как оба операнда функции, это - беззнаковые, целые числа с разрядностью 8 бит, то максимально число, которое можно получить на выходе – $255 \cdot 255 + 255^3 = 16646400$, данное число можно уместить, используя 24 бита, тогда выходной сигнал будет иметь разрядность в 24 бита.

- Разработанный блок умножения, будет умножать 8ми и 16ти разрядные числа.
- Регистр А имеет 8 разрядов
- Регистр В имеет 16 разрядов
- Регистр АВ имеет 16 разрядов (т.к. хранит результат умножения двух 8 битовых чисел)
- Регистр ААА имеет 24 разряда (т.к. хранит результат умножения трех 8 битовых чисел)

Результат тестирования разработанного блока.

Пример тестового окружения для разработанной функции.

```
task test_func;
    input [3:0] iter;
    input [7:0] test_a;
    input [7:0] test_b;
    input [23:0] test_res;
    begin
        a = test_a;
        b = test_b;
        start = 1;
        #10
        start = 0;
        while (busy) begin
            #5;
        end
        if(y == test_res) $display("Test %d: Correct", iter);
        else $display("Test %d: Failed %d", iter, y);
    end
endtask
```

Рисунок 5 - код тестовой функции

```
// 10 test
test_func(1, 0, 0, 0);
test_func(2, 1, 0, 1);
test_func(3, 1, 1, 2);
test_func(4, 2, 2, 12);
test_func(5, 15, 240, 6975);
test_func(6, 240, 15, 13827600);
test_func(7, 170, 85, 4927450);
test_func(8, 85, 170, 628575);
test_func(9, 100, 125, 1012500);
test_func(10, 255, 255, 16646400);
```

Рисунок 6 - Тестовые случаи

Временная диаграмма тестирования.

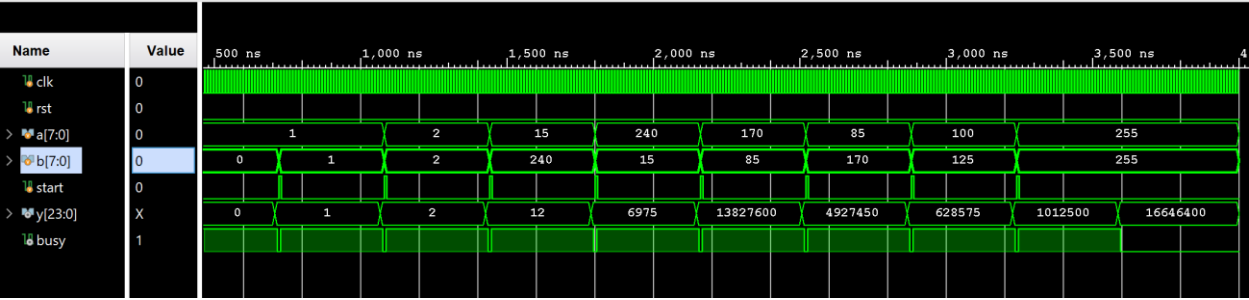


Рисунок 7 - временная диаграмма тестирования

Вывод в консоль:

```
Test 1: Correct
Test 2: Correct
Test 3: Correct
Test 4: Correct
Test 5: Correct
Test 6: Correct
Test 7: Correct
Test 8: Correct
Test 9: Correct
Test 10: Correct
```

Рисунок 8 - вывод в консоль

Время вычисления результата при частоте тактового сигнала в 100 МГц.

Тактовая частота в 100МГц – это 1 синхроимпульс за 10нс. Временная диаграмма тестирования для самых больших входных чисел из ОДЗ:

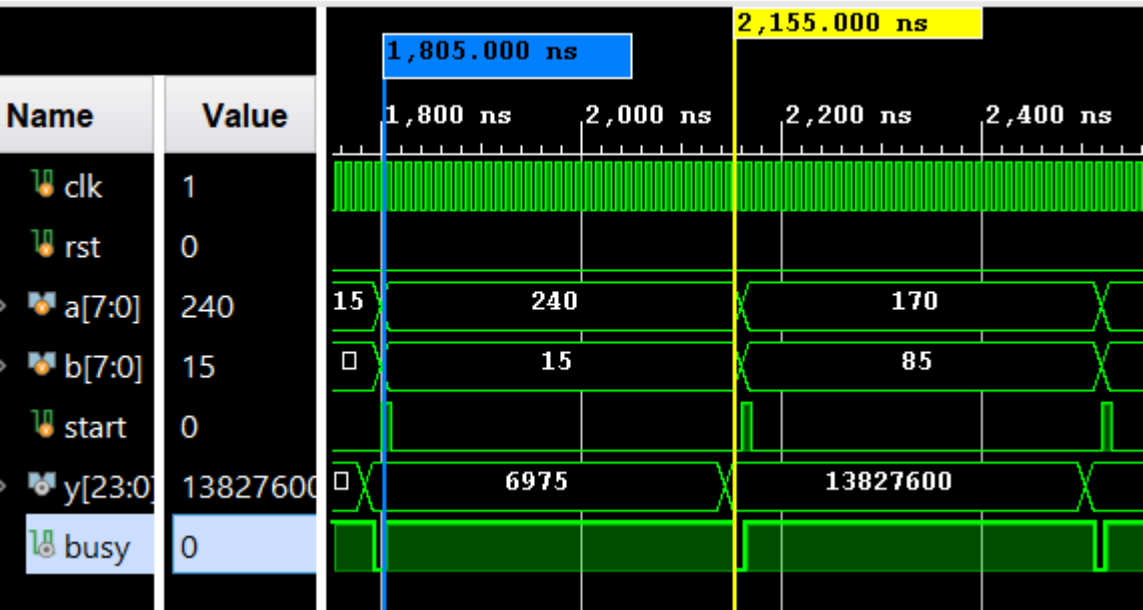


Рисунок 9 - 100 МГц

По скриншоту временной диаграммы видно, что на вычисление результата функции при тактовой частоте 100МГц потребовалось 350нс.

Выводы.

Во время выполнения данной лабораторной работы я разработал блок вычисления функции и реализовал его на языке описания аппаратуры Verilog, для его реализации был применен автомат Мура и составлен конечный автомат. Изучил различные аспекты реализации последовательной логики.